SONIFICATION SANDBOX RECONSTRUCTION: SOFTWARE STANDARD FOR AUDITORY GRAPHS

Benjamin K. Davison and Bruce N. Walker

Sonification Lab, School of Psychology
Georgia Institute of Technology
Atlanta, GA 30332
ben@cc.gatech.edu, bruce.walker@psych.gatech.edu

ABSTRACT

We report on an overhaul to the *Sonification Sandbox*. The *Sonification Sandbox* provides a cross-platform, flexible tool for converting tabular information into a descriptive auditory graph. It is implemented in Java, using the Java Sound API to generate MIDI output. An improved modular code structure provides a strong user interface and model framework for auditory graph representation and manipulation. A researcher can integrate part or the entire program into a different experimental implementation. The upgraded *Sonification Sandbox* provides a rich description of the auditory graph representation that can be saved or exported into various file formats. This description includes data representations of pitch, timbre, polarity, pan, and volume, along with graph contexts analogous to visual graph axes. Applications for the *Sonification Sandbox* include experimentation with various sonification techniques, data analytics beyond visualization, science education, auditory display for the blind, and musical interpretation of data.

[Keywords: Sonification Sandbox, auditory graphs, Java sound]

1. INTRODUCTION

Sonification is the use of non-speech audio to convey information. Auditory graphs are the sonification parallel to visual graphs. Walker and Nees argue that auditory graphs research and development is lacking a standard, shared model for communication and implementation [1]. The present paper outlines major changes to the *Sonification Sandbox* to represent the developing graph model and to provide easy usage as a standalone tool or software library for new auditory graph programs.

The *Sonification Sandbox* is a multipurpose system which integrates contemporary auditory graphing knowledge with a modular software program [2,3]. The application has now been completely upgraded in order to address previous shortcomings, add new features, and enable a more robust and extensible platform. In particular, the *Sonification Sandbox* now features a data model that parallels current academic understanding of auditory graphs components. The software additionally offers a user interface, model, and translation structure which provides an auditory graph toolkit out of the box and a software library for experimental research programs.

The *Sonification Sandbox* remains a cross-platform, multipurpose implementation. The Java program can be run on many PC and Macintosh hardware and operating system configurations. It is intended for use for science education, auditory display research and development, and auditory display for the blind.

The current efforts in the *Sonification Sandbox* represent a fundamental shift from the original proof-of-concept implementation [2,3] into a descriptive and flexible tool for researchers, developers, and end users. The software is intended to be available as a standalone program or as a software object library in a larger system. The *Sonification Sandbox* can also be used as a translator between various auditory graph file formats including the new data model.

2. RELATED WORK

The *Sonification Sandbox* was first developed as a general-purpose solution for creating auditory graphs [2,3]. Its original implementation was inspired heavily by Lodha's Listen, Muse, and MUSAR [4,5,6] and Upson's Sound Grid [7]. However, even in its original form the *Sonification Sandbox* exhibited a greater range of features than many other systems. For example, from the outset the previous versions of the software have provided a graphical toolkit for a variety of user domains, including researchers, teachers, and musicians. It supplied a user interface that affords the contextual framework required for a useful auditory graph. It expanded the output of auditory graphs to include information such as context and mappings. The *Sonification Sandbox* also allowed the user to explore and interact with the auditory graph: looping, pausing, and holding the graph allowed the user to focus on a particular part, much like a sighted user can visually examine a particular area of a graph. Finally, the *Sonification Sandbox* provides a cross-platform Java solution.

However, the original *Sonification Sandbox* was severely limited in its data representation. While the user interface supplied the necessary options, the backend essentially manipulated an array of data values unconnected from the mappings and context related to the data. This approach limited the user to saving only the data's original values in comma separated values (CSV, a format for tabular data) or exporting everything into MIDI. In addition, the centralized structure of the software code was inflexible to anticipated upgrades.

The *Sonification Sandbox* relies on open source libraries for portions of the system not directly related to auditory graph interpretation. The major systems are the data table (currently implemented via JTable) and a visual graph (implemented via JFreeChart).

3. CURRENT DESIGN

The most recent *Sonification Sandbox* stable and development releases are online at http://sonify.psych.gatech.edu/research. The program is wrapped into a cross-platform installer. The software requires Java Runtime Environment (JRE) version 1.5.0 or higher.

The new *Sonification Sandbox* significantly changed our approach to representing information in an auditory graph. First, our system features a data model that more closely matches the theoretical frameworks being developed for auditory graphs, as in [8].

Second, the *Sonification Sandbox* is now a modular system which can be integrated into other systems as a complete package or in parts.

3.1. Data Model

Auditory graphs contain a rich amount of information beyond a sequence of frequencies. Data Mappings enrich the depth of information played. A series of tones has a variety of important properties. Pitch range affects the note change between two different data points. Humans can more easily detect a large change. Timbre, pan, and volume differences can help a listener distinguish between two simultaneous sequences of data points. Pitch polarity can help the listener understand the meaning of an increasing or decreasing function.

Context provides information related to the position of the notes with a fixed line. Like visual graphs, auditory graphs require references for interpreting a sound. Similar to the visual counterpart, the *Sonification Sandbox* has the concept of x and y axes. While the context and mappings ideas were present in previous versions of the software, the data model now accurately captures their conceptual relationship with the data series.

Figure 1 illustrates a simplified UML diagram of the software's auditory graph model. An Auditory Graph class is designed to contain all the information necessary to display an auditory graph on any user interface implementation. The Auditory Graph Context represents information related to context in the auditory graph. On a visual graph, this corresponds to the horizontal and vertical axes, their tick marks, and labels [9, 1]. An Auditory Channel is a representation of information related to a specific thread of information characterized on the graph. A thread of information corresponds to a particular row or column on a table or a line on a visual graph. The Auditory Channel can provide statistical measures of the channel's information, such as minimum, maximum, and median values. This information is useful for creating meaning for contexts. Channel Mappings provides a rich level of information related to the channel, including pan, pitch, volume, polarity, and timbre. The Sound Point serves as the storage location for the single value of the data point. In the previous versions of the *Sonification Sandbox*, described in [2,3], the Sound Point's pitch was essentially the only item which could be saved for reuse between program sessions. The new *Sonification Sandbox* also integrates implementation-specific elements, such as a Project concept which can hold many graphs, into the model. Such changes are driven by software usability considerations beyond the scope of auditory graphs research.

The auditory graph model is intended to develop as the sonification community molds the theoretical graph model. The *Sonification Sandbox* will attempt to model the most contemporary understanding of auditory graphs. Anyone with the source code could easily extend the auditory graph model to include new ideas in their own experiments.

Our auditory graph model is fully Serializable: the model can be saved directly into a format that retains the class structure for Java. This approach has two advantages over implementations in file formats such as CSV, MIDI, or XML. Data saving and opening does not need to undergo any format transformation process. File format translation can be time consuming and may introduce errors or data loss; these problems are avoided with Serialization. Second, built-in file compression provides file sizes smaller than uncompressed formats. The native *Sonification Sandbox* file format has an ".ssp" (*Sonification Sandbox* Project) extension and can be shared between colleagues and operating systems. Along with the program-specific format, the *Sonification Sandbox* will currently import CSV files and export in CSV and MIDI formats.

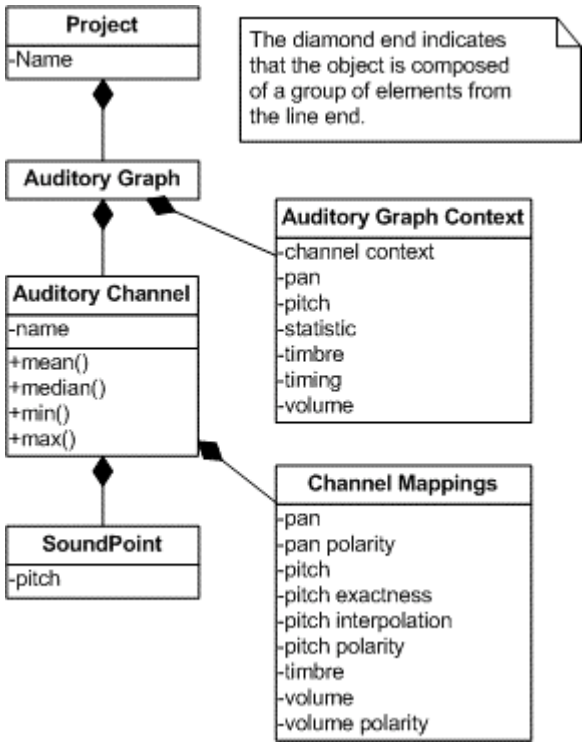

Figure 1. *The Auditory Graph data model (UML). The model provides a rich representation of graph data.*

3.2. Modular Subsystems

The *Sonification Sandbox* was redesigned with modularity in mind. All UI, translation, and model components can be used outside of the environment implemented within the *Sonification Sandbox*. Figure 2 displays the overall relationships between the different parts of the *Sonification Sandbox*. The user interface acts as a medium between the user and the auditory graph model. Translators take the data model and prepare the auditory graph for export into various file formats. Like the auditory graph model, each major module has a set of interrelated submodules. Someone unsatisfied with the *Sonification Sandbox* may extend

its abilities in one area while relying on the software's abilities in others.

For example, a researcher may want to develop an experiment that only explores the effects of changes in panning on a single channel of information. That researcher could implement a system that uses the Channel, Channel Mappings, and Sound Point section of the auditory graph model, along with the corresponding user interfaces. This system could have its own sound interpreter and player, such as a pure tone player instead of the *Sonification Sandbox*'s MIDI player. Instead of spending considerable programming time developing existing components, the researcher can focus on completing an experiment.

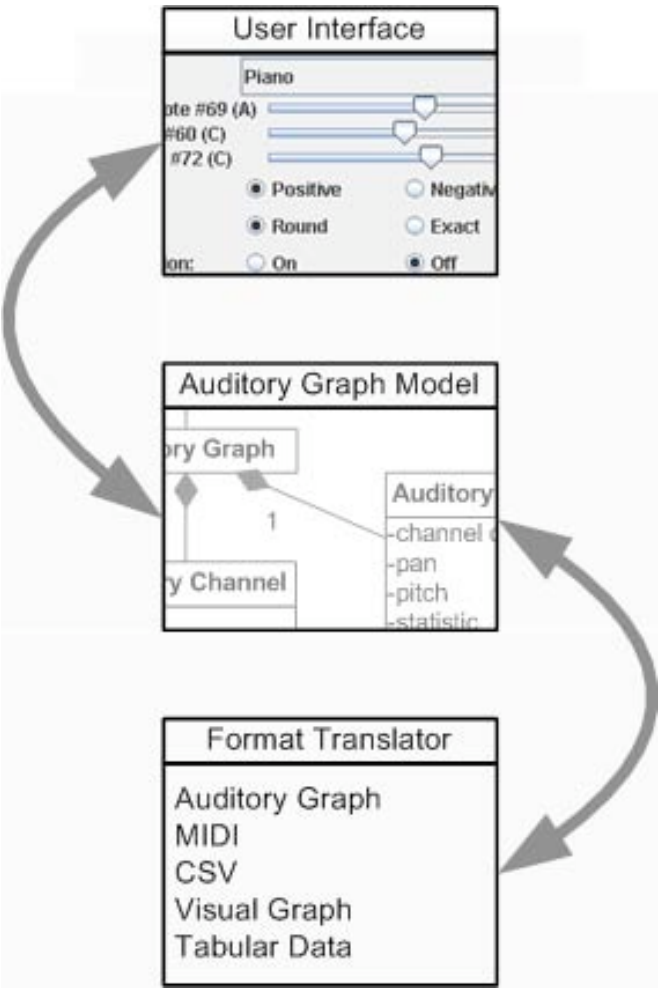

Figure 2. *The Sonification Sandbox has three main modules. The user interface informs the auditory graph model of user changes. Translators take the model and turn it into other representations.*

3.3. Other Modifications

Along with the changes described above, there are a variety of minor modifications. The Java MIDI Synthesizer is now provided with the *Sonification Sandbox*. A MIDI synthesizer is a list of records describing how to play certain MIDI instruments. In previous implementations, two separate computer systems could have rendered a given MIDI file (i.e., an auditory graph saved out from the *Sonification Sandbox*) in noticeably different ways. For example, one system could have used a piano to play a given data stream, whereas the other could have used a violin, depending on the exact differences between the two MIDI setups or synthesizers installed. Including the Java Synthesizer helps ensure all renditions of an auditory graph sound the same, regardless of platform. This increased commonality enhances collaboration and shared investigation.

In addition, mappings and context settings will play audio feedback when changed. The feedback tones relate to the changed setting, such as a specified volume after a volume slider is released. The major modules and some submodules are threaded to enhance performance and usability. Finally, all external components are updated to the most recent software version.

3.4. Ongoing Development

The *Sonification Sandbox* is, once more, an active project undergoing regular updates, enhancements, and extensions. After our current release, we intend to iterate additional features supporting the new system.

As one example, the *Sonification Sandbox* is currently limited to a single MIDI synthesizer. In some applications, a non-default synthesizer may be desired. We would like the user to be able to add and distribute custom synthesizers. The synthesizer would have to be created or imported from outside of the *Sonification Sandbox*.

While modularity is available for the current system, we intend to improve the programming usability of the classes. This will be an iterative process, building on software analysis and user feedback. JavaDoc documentation is built into the development process to facilitate class and method understanding.

The auditory graph model fully describes our target production. Translators provide tools for conversion into MIDI and CSV. However, these may not be as portable as other display formats such as QuickTime or MP3. XML could be used to communicate the meaning of a graph with other auditory graphing software. We intend to create translators within the *Sonification Sandbox* to support exportation into these formats. The *Sonification Sandbox* will also be able to import data from more formats beyond .ssp and CSV files.

We intend to expand the accessibility of our software. The *Sonification Sandbox* is accessible to users with vision impairments using third-party software such as the Java Accessibility Bridge [alluded to in 6], but we have not yet directly supported fully accessible usage. While the auditory graph end product is intended for use by both blind and sighted audiences, a visually impaired *Sonification Sandbox* user may still require sighted assistance to complete complicated tasks. To the extent possible, we will be working to eliminate such issues.

A command-line tool could produce output for simple operations such as CSV conversion to MIDI. Such an action would require little user interface beyond a command help system. It would facilitate *Sonification Sandbox* tool utilization as a batch job or component of another system. It may also be more appropriate for human interaction in cases when visual display is limited.

Finally, we intend to extend the functionality of the data table. Many graphs characterize mathematical functions. Spreadsheet applications such as Microsoft Excel provide intuitive layers separating formula and value. Without this tool, users may be forced to use a spreadsheet application to export value-only data any time there is a data change. As a more distant goal, we hope to support streaming information such as RSS feeds of weather data.

3.5. System Scope

The *Sonification Sandbox* has limitations, some by design and some a function of the architecture or software. General MIDI restricts playback of data to 16 streams of information. This includes all data mappings, contexts, and data channels. Many graphs can be created by the *Sonification Sandbox*. However, the *Sonification Sandbox* is not intended to support every possible implementation of the sonification of visual graphs. Instead, it is designed to provide a general-purpose software tool to build auditory graphs. Furthermore, its tested model, translation algorithms, and user interface can be modified to create new programs that provide greater functionality.

4. CONCLUSION

The original *Sonification Sandbox* provided a baseline system for creating auditory graphs in many domains. The new *Sonification Sandbox* has been fully rebuilt so that it may be integrated in part or entirely into other systems on a wide variety of PC and Macintosh platforms. In addition, the data is now represented in a way which represents the current theory in auditory graphs research.

5. REFERENCES

[1] Walker, B.N. and Nees, M.A., "An agenda for research and development of multimodal graphs," in *Proceedings of the International Conference on Auditory Display*. 2005. Limerick, Ireland.

[2] Walker, B.N. and Cothran, J.T., "*Sonification Sandbox*: a graphical toolkit for auditory graphs," in *Proceedings of the International Conference on Auditory Display*. 2003. Boston, USA.

[3] Walker, B.N. and Lowey, M., "*Sonification Sandbox*: A graphical toolkit for auditory graphs," presented at Rehabilitation Engineering & Assistive Technology Society of America (RESNA) 27th International Conference, Orlando, FL, 2004.

[4] Wilson, C.M. and Lodha, S.K., "Listen: A data sonification toolkit," in *Proceedings of the International Conference on Auditory Display*. 1996. Palo Alto, USA.

[5] Lodha, S.K., Beahan, J., Heppe, T., Joseph, A.J., and Zne-Ulman, B., "MUSE: A musical data sonification toolkit," in *Proceedings of the International Conference on Auditory Display*. 1997. Palo Alto, USA.

[6] Joseph, A.J. and Lodha, S.K., "MUSART: Musical Audio Transfer Function Real-Time Toolkit," in *Proceedings of the International Conference on Auditory Display*. 2002. Kyoto, Japan.

[7] Upson, Robert. "Educational sonification exercises: Pathways for mathematics and musical achievement," in *Proceedings of the International Conference on Auditory Display*. 2002. Kyoto, Japan.

[8] Nees, M.A. and Walker, B.N., "Listener, task and auditory graph: toward a conceptual model of auditory graph comprehension," in *Proceedings of the International Conference on Auditory Display*. 2007. Montréal, Canada.

[9] Smith, D.R. and Walker, B.N., "Tick-Marks, axes, and labels: The effect of adding context to auditory graphs," in *Proceedings of the International Conference on Auditory Display*. 2002. Kyoto, Japan.

DOWNLOAD INFORMATION

The Sonification Sandbox can be downloaded at: http://sonify.psych.gatech.edu/research